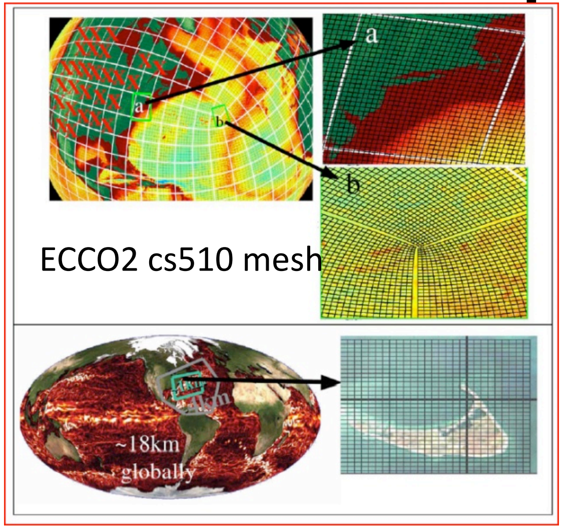
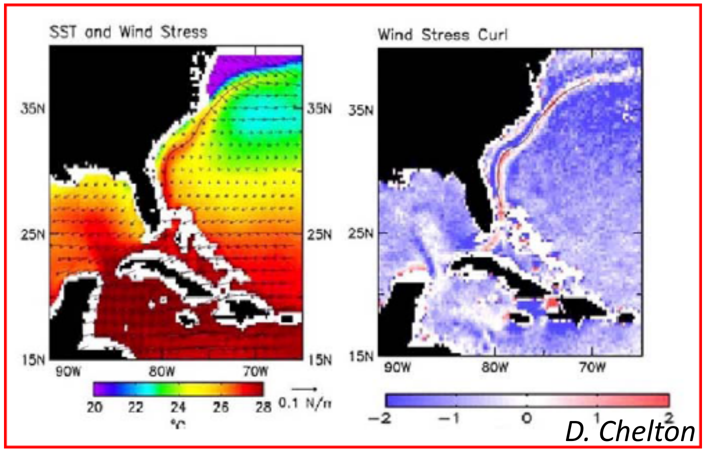
High-resolution grids and coupling/regridding.



ECCO2 cs510 mesh



D. Chelton

Chris Hill, cnh@mit.edu

ECCO2 Sept. 2008

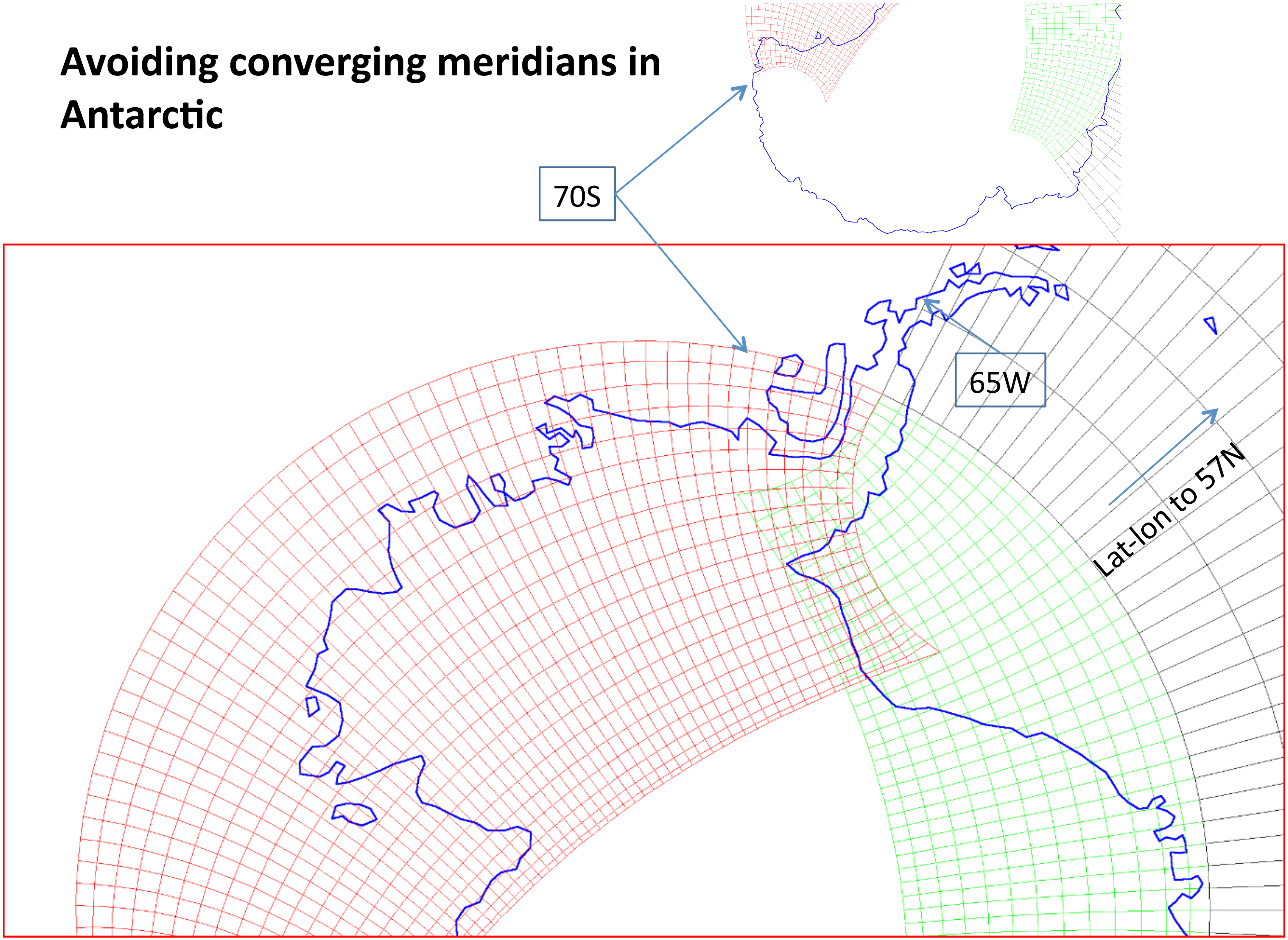
High-res horizontal grid strategy

- Targeting base grid at very high-resolution that can be "cleanly" coarsened (and can be used globally or regionally).
  - ➔Largest number of common integer factors – (degrees, minutes, seconds are really useful)

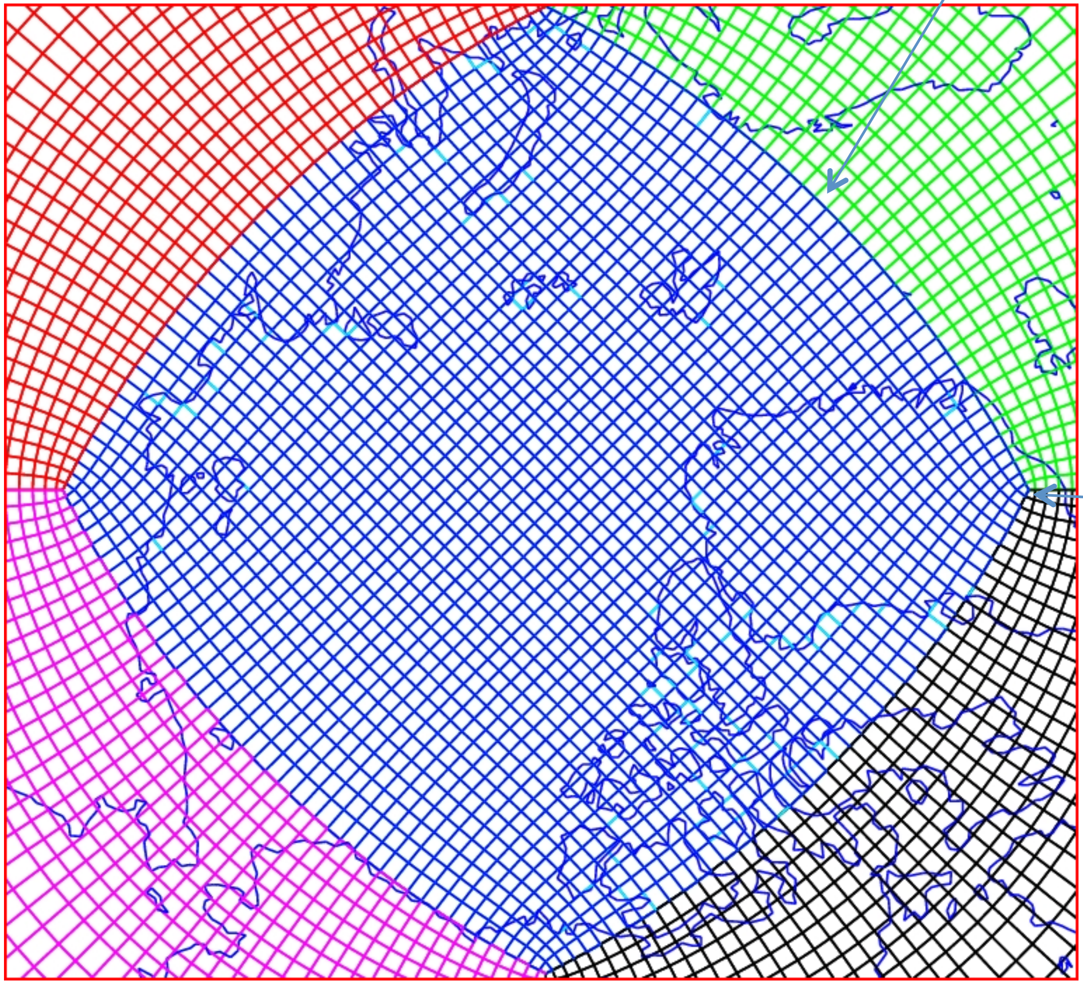
    510 – 16 factors, 512 – 10 factors, 360 – 24 factors, 4320 – 48 factors.
  - ➔Grid spacing should reflect deformation radius. Highest res should resolve e.g. Δ < 2km @ $f\sim10^{-4}s^{-1}$.
  - ➔Better treatment of polar regions, quasi-isotropic grid spacing in high-latitudes – next slides.

- Sticking with orthogonal curvilinear grid?
  - High-order numerics cleaner. OS7MP etc…
  - Grid generation more challenging.
  - For now, targeting orthogonal, but maybe should look more closely at trade offs (cf Putnam and Lin).
    - Numerical generation is (much) harder
    - Grid can be hard to control
    - But we think we understand numerics

Avoiding converging meridians in Antarctic

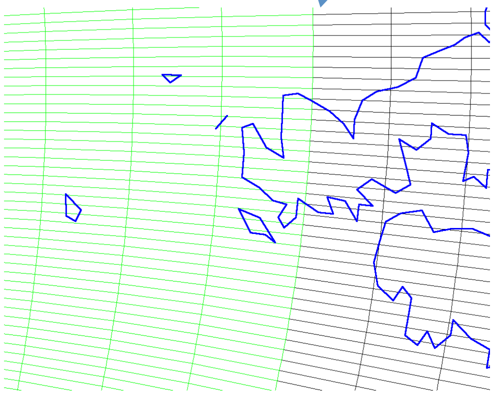


Compatible Polar Cap “Isotropic”
Arctic



~63N

38W, 67.5N



57N

Lat-lon to 70S

Compatible grid hierarchy

Requirements – large number of factors **and** grid lines at 65W, 38W, 38W+90, 38W+180, 38W+270 ➔ 18 levels of refinement ranging from 17280 – 80 points around globe.

	Δ_{arctic}	Δ_{eq}	**No. horiz cells**
1/48	943m	2.3km	141×10^6
1/24	1.9km	4.6km	35×10^6
1/16	2.8km	6.9km	15×10^6
1/12	3.8km	9.3km	9×10^6
1/8	5.6km	13.9km	4×10^6
1/6	7.5km	18.5km	2×10^6
3/16	8.5km	20.8km	1.7×10^6
¼	11.3km	27.8km	1×10^6
1/3	15.1km	37.1km	5×10^5
3/8	17.0km	41.7km	4×10^5
½	22.6km	55.6km	2.5×10^5
9/16	25.5km	62.5km	2×10^5
¾	34.0km	83.4km	1×10^5
1	43.3km	111.2km	6×10^4
1.125	51.0km	125.1km	5×10^4
1.5	68.0km	166.8km	3×10^4
3	135.9km	333.5km	7×10^3
4.5	203.8km	500.3km	3×10^3

…. all we need is a foundation, general curvilinear 17280 point polar-cap and Antarctic split grid ☺

Generating high-res Arctic and Antarctic grids.

- Two existing grid generation tools for MITgcm. Initial plan was to use these, however:
  - Standard cube-sphere generating code (J Purser code) based on hard coded, high-order numerical solution to analytic conformal map (angle preserving) of unit circle to unit square. Not good for general grid generation, but very accurate.
  - Spgrid (E Hill code) solves elliptic problem on a sphere for a somewhat irregular domain, but
    - Has accuracy issues for large problem sizes.
    - Region boundaries cannot be arbitrary curves. Precludes "recursive" approach to handling large problem size.
    - Has occasional orthogonality issues.
    - Coded for sphere.

- Neither of two approaches can create accurate grids with complete flexibility.
  - New tool based on "conformal mapping" of arbitrary two-dimensional shape outline to a rectangle. Fairly standard 2-d method, can be implemented with 4th order accuracy. (e.g. ROMS uses 2-d, 2nd order accurate formulation from Aero community [Wilkin; Ives and Zacharias]).
  - To accommodate true spherical regions (i.e. 3d surface) added additional polar stereographic projection of region from three to two dimensions (this is a conformal map).
  - Together this allows arbitrary boundaries on sphere, so can
    - solve for meshes recursively ➔ large problems can be solved in pieces.
    - make arbitrary cuts

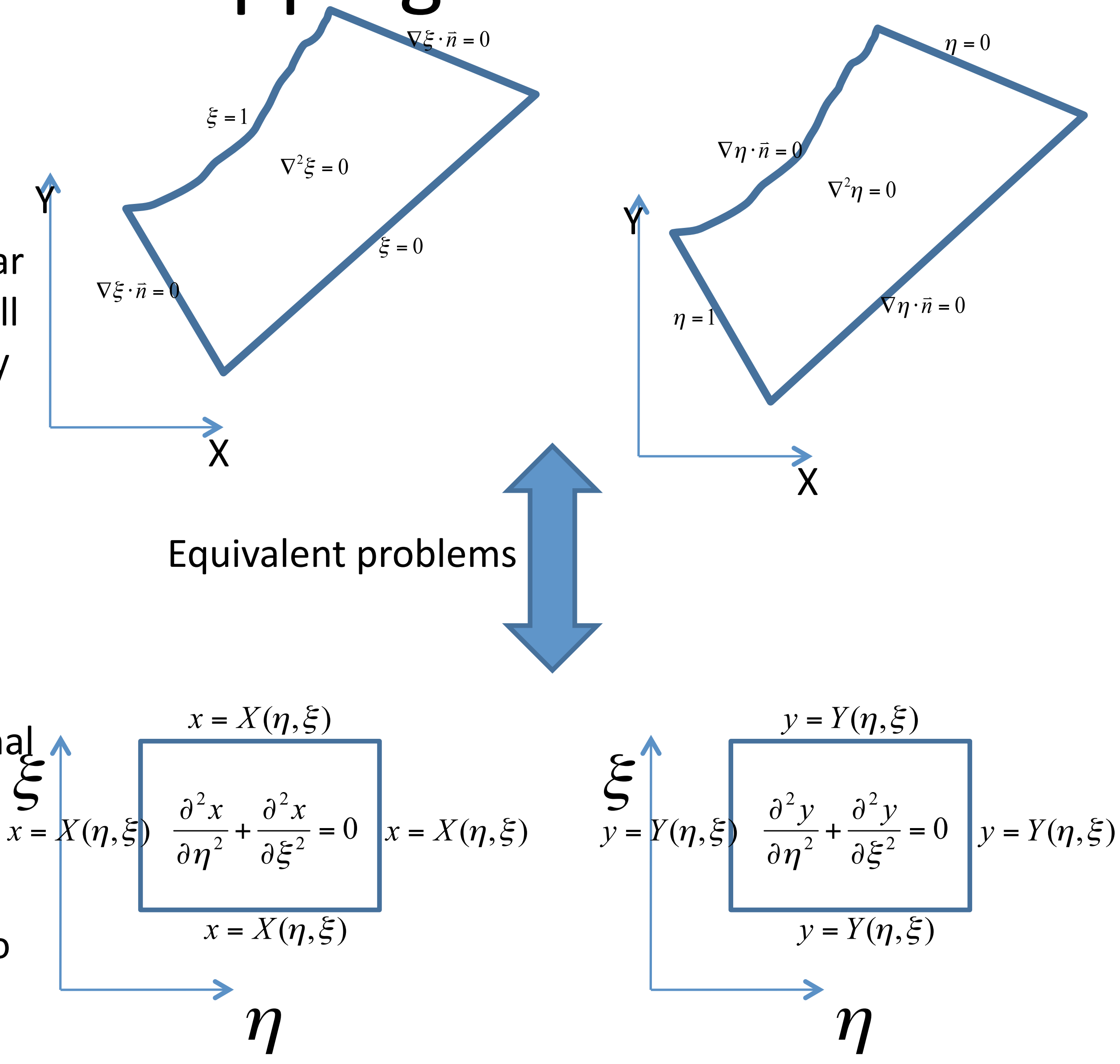
Grid generation with conformal mapping

Method 1

X,Y coordinates. Solve elliptic problem for η and ξ in irregular domain, discretization not well defined. Hard to do accurately and then recover x and y.

Method 2

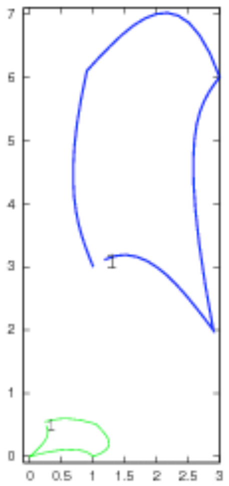
η, ξ coordinates. Solve elliptic problem for x and y in regular domain. Boundary Dirichlet values deduced from conformal transformations (X=X_1 o X_2 etc…) that preserve angles in rectangle interior.
Well established in Aero/Astro e.g. Trefethen SCPACK, MIT 1989

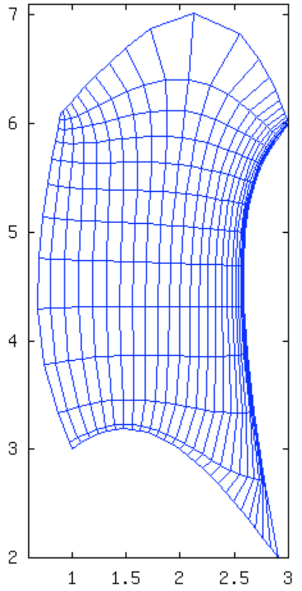

$\nabla\xi\cdot\vec{n}=0$
$\xi=1$
$\nabla^2\xi=0$
$\xi=0$
$\nabla\xi\cdot\vec{n}=0$
Y
X
$\eta=0$
$\nabla\eta\cdot\vec{n}=0$
$\nabla^2\eta=0$
$\eta=1$
$\nabla\eta\cdot\vec{n}=0$
Y
X
Equivalent problems
$x=X(\eta,\xi)$
ξ
$x=X(\eta,\xi)$
$\frac{\partial^2 x}{\partial\eta^2}+\frac{\partial^2 x}{\partial\xi^2}=0$
$x=X(\eta,\xi)$
$x=X(\eta,\xi)$
η
$y=Y(\eta,\xi)$
ξ
$y=Y(\eta,\xi)$
$\frac{\partial^2 y}{\partial\eta^2}+\frac{\partial^2 y}{\partial\xi^2}=0$
$y=Y(\eta,\xi)$
$y=Y(\eta,\xi)$
η

Steps for grid generation using **Method 2**

- Define four arbitrary edges on sphere (each edge is a sequence of lat-lon coordinates).
- Project edges onto 2d plane using polar stereographic projection (a conformal map).
- Iteratively apply “hinge transformation” (a standard conformal map function) to edge points in turn until resulting shape is rectangular in η and ξ.
- Solve regular elliptic probs for x and y locations at which η and ξ lines cross.
- Project back onto sphere using inverse polar stereographic projection.

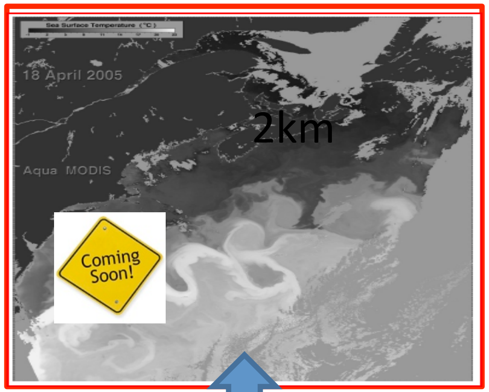
e.g.

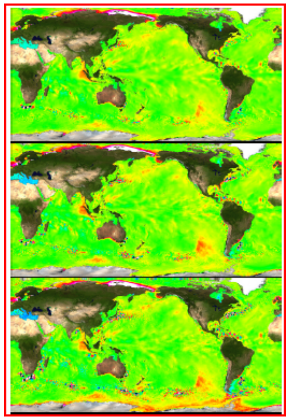


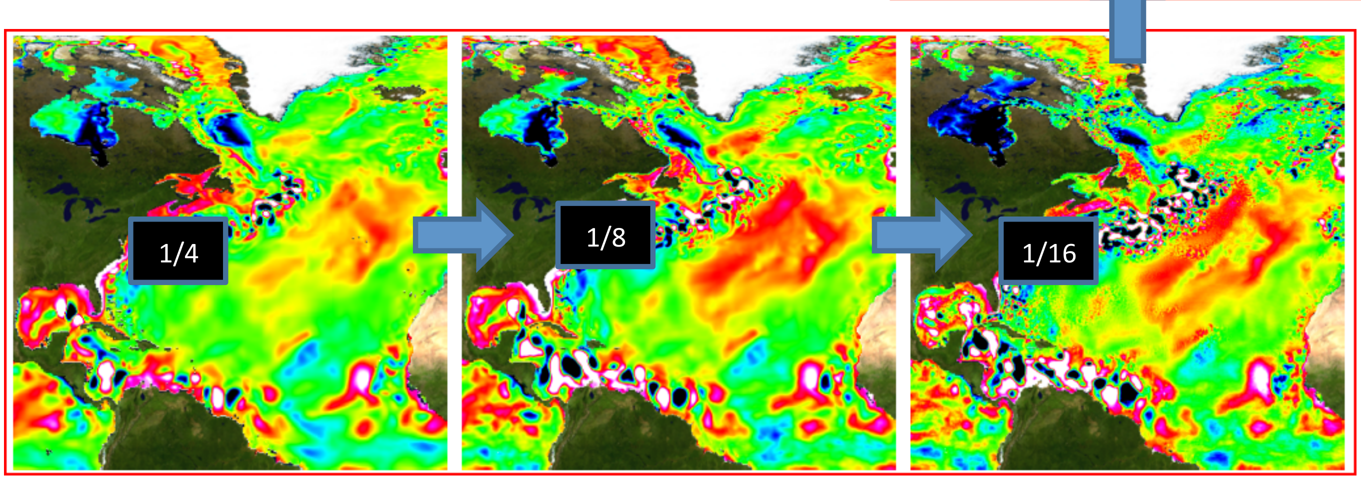


Summary

- Can generate cube, cap and arbitrary meshes using Octave/Matlab code
  - Currently validating symmetry of assembled (multi-panel) mesh, simple rotation tests, cube sphere etc…
  - Significantly improved accuracy in mesh (and efficiency) for recursive, high-resolution solutions.
  - Fourth order elliptic solver needs coding.







Coupling

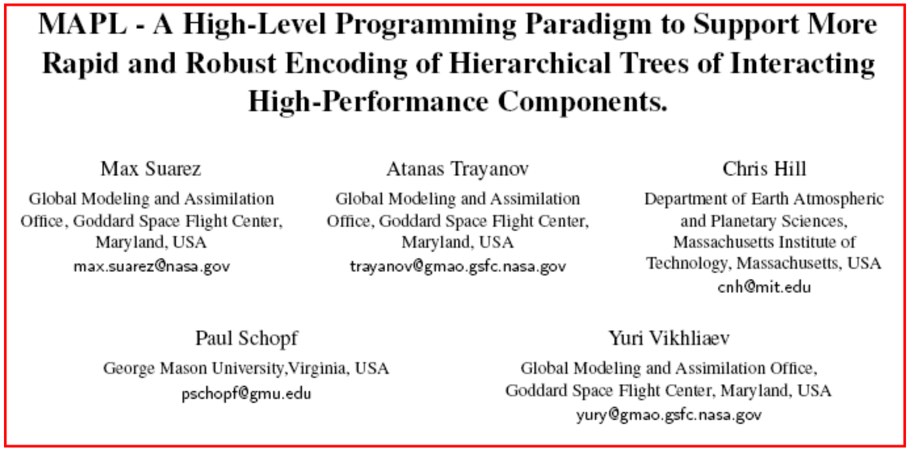
- Work originates in bringing MIT ocean, sea-ice, ocean eco and Goddard GEOS-5 system together.
- Work produced a general (but formal) foundation mechanics for a variety of complex “coupling” scenarios e.g. multi-scale/embedded ←→coupled atmos-ocean.
- Two areas
  - MAPL interfaces
  - Regridding and multi-scale/ multi-component modeling

- Move away from a monolithic modeling system perspective to more of a “software ecosystem”.

MAPL

- Thin layer built on top of ESMF that makes it possible for MIT (me), GFDL (Balaji, Niki Zadeh), GMU (Paul Schopf) and NASA GMAO (Max Suarez, Atanas Trayanov) to interchange code!

- MAPL stands for MAP layer.
- Described in



MAPL - A High-Level Programming Paradigm to Support More Rapid and Robust Encoding of Hierarchical Trees of Interacting High-Performance Components.

Max Suarez
Global Modeling and Assimilation Office, Goddard Space Flight Center, Maryland, USA
max.suarez@nasa.gov

Atanas Trayanov
Global Modeling and Assimilation Office, Goddard Space Flight Center, Maryland, USA
trayanov@gmao.gsfc.nasa.gov

Chris Hill
Department of Earth Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Massachusetts, USA
cnh@mit.edu

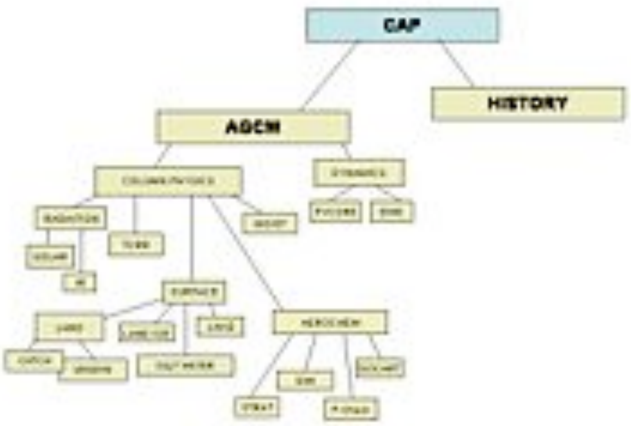
Paul Schopf
George Mason University,Virginia, USA
pschopf@gmu.edu

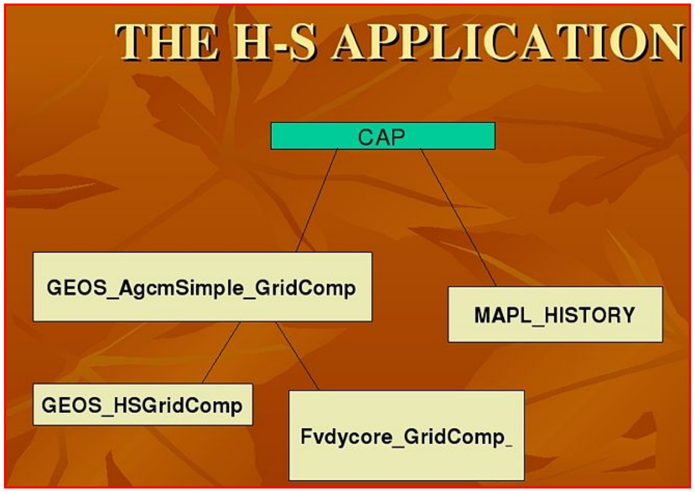
Yuri Vikhliaev
Global Modeling and Assimilation Office, Goddard Space Flight Center, Maryland, USA
yury@gmao.gsfc.nasa.gov

Also - http://maplcode.org/maplwiki/index.php

Best way to learn about MAPL.

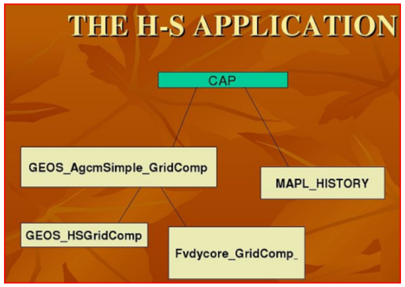
MAPL abstraction




THE H-S APPLICATION

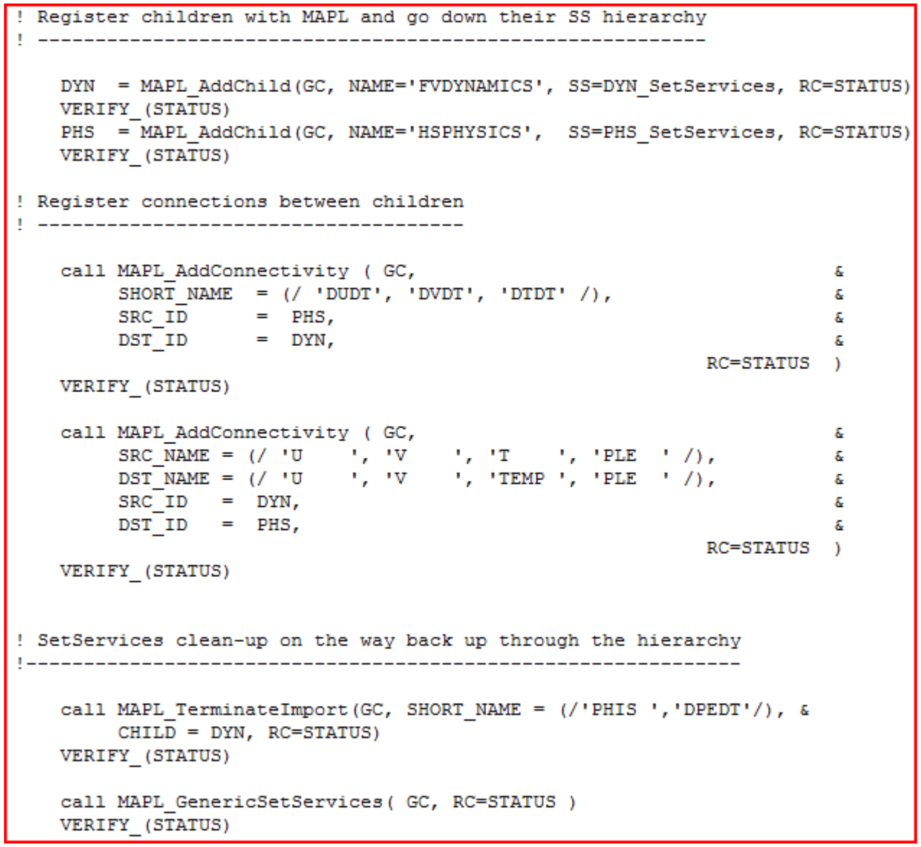
- In MAPL an application is a graph (or tree) of components (these contain the program logic).
- System sets rules designed to make it “easy” to cut, replace, share assemble and reassemble components trees as appropriate.
  - Data flows up and down branches only and is always transported in standard, labeled containers (defined by ESMF).
  - All components have a defined set of actions they support and understand.
- Rules make it possible to trigger life-cycle steps (boot-up, get ready, go, stop etc..) over an entire tree automatically in an OO style.

MAPL in action - SetServices



SetServices is first stage in life-cycle.

- 1 – CAP calls `MAPL_CAP`
  - Sets up HISTORY (always)
  - And calls other child **`SetServices`**

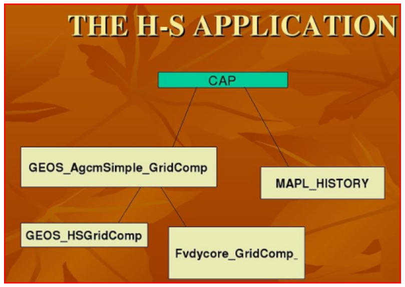


```
! Register children with MAPL and go down their SS hierarchy
! ---------------------------------------------------------

    DYN  = MAPL_AddChild(GC, NAME='FVDYNAMICS', SS=DYN_SetServices, RC=STATUS)
    VERIFY_(STATUS)
    PHS  = MAPL_AddChild(GC, NAME='HSPHYSICS',  SS=PHS_SetServices, RC=STATUS)
    VERIFY_(STATUS)

! Register connections between children
! ------------------------------------

    call MAPL_AddConnectivity ( GC,                                    &
         SHORT_NAME  = (/ 'DUDT', 'DVDT', 'DTDT' /),                   &
         SRC_ID      =  PHS,                                           &
         DST_ID      =  DYN,                                           &
                                                            RC=STATUS  )
    VERIFY_(STATUS)

    call MAPL_AddConnectivity ( GC,                                    &
         SRC_NAME = (/ 'U    ', 'V    ', 'T    ', 'PLE  ' /),          &
         DST_NAME = (/ 'U    ', 'V    ', 'TEMP ', 'PLE  ' /),          &
         SRC_ID   =  DYN,                                              &
         DST_ID   =  PHS,                                              &
                                                            RC=STATUS  )
    VERIFY_(STATUS)


! SetServices clean-up on the way back up through the hierarchy
!--------------------------------------------------------------

    call MAPL_TerminateImport(GC, SHORT_NAME = (/'PHIS ','DPEDT'/), &
         CHILD = DYN, RC=STATUS)
    VERIFY_(STATUS)

    call MAPL_GenericSetServices( GC, RC=STATUS )
    VERIFY_(STATUS)
```

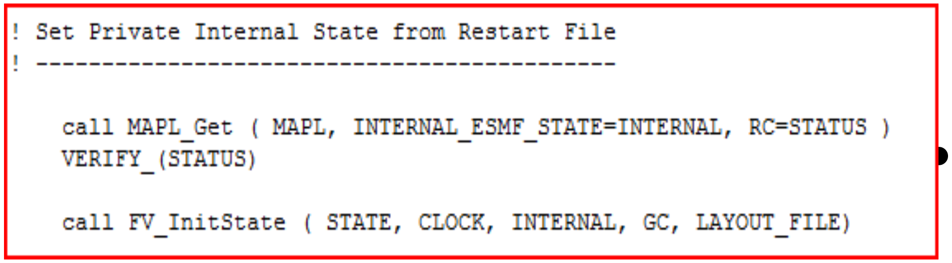
2– AGCM `SetServices` registers its children and the data flows into and out of them and the data flows between them.

3– the tree is traversed recursively until last node and then control returns to CAP that call the `Initialize` phase.

MAPL in action - Initialize



Initialize is second stage in life-cycle.

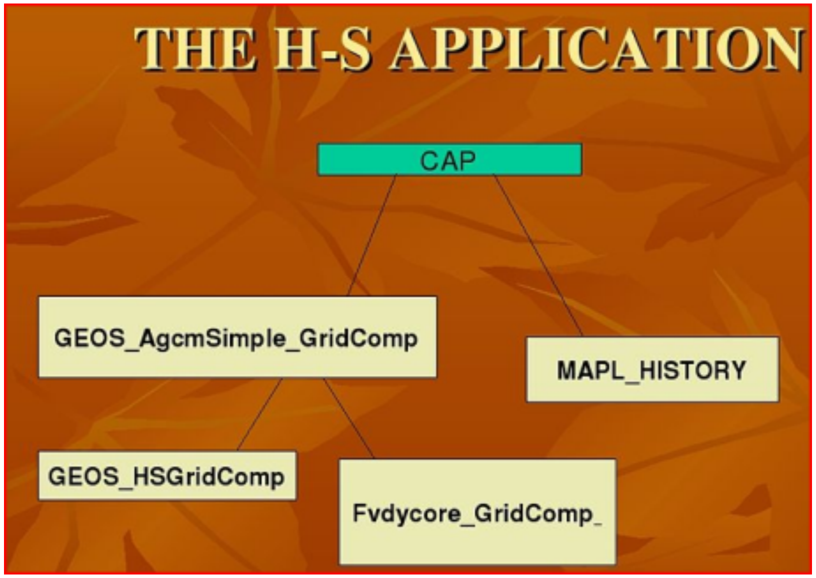


```
! Set Private Internal State from Restart File
! --------------------------------------------

    call MAPL_Get ( MAPL, INTERNAL_ESMF_STATE=INTERNAL, RC=STATUS )
    VERIFY_(STATUS)

    call FV_InitState ( STATE, CLOCK, INTERNAL, GC, LAYOUT_FILE)
```

- AGCM `Initialize` called from CAP.
- AGCM automatically calls child `Initialize` routine.
- Each child fills some internal data structures with initial values that are then stored with the component.

MAPL in action – example overall lifecycle.

Example Overall lifecycle



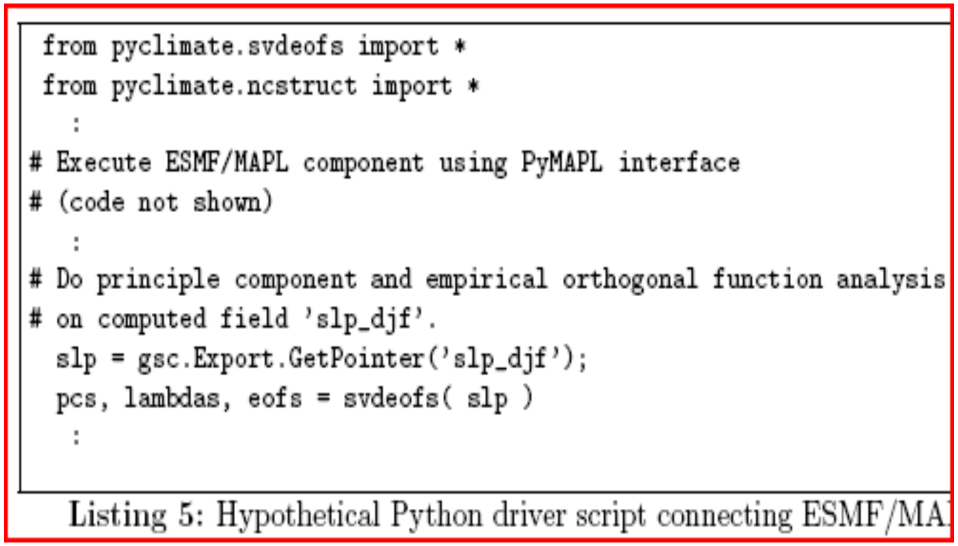
- SetServices ➔ Initialize ➔ Run ➔ Checkpoint ➔ Finalize
- Ticked from top during each lifecycle phase.
- Tree is traversed recursively at each tick.
- Example available at

http://maplcode.org

MAPL Summary

- MAPL defining a formal "architecture" for coupled Earth System models..
  - Architecture separates wiring of coupled pieces cleanly but very flexibly.
  - Logic at core of coupled pieces can remain unchanged.
  - However you have to use MAPL for wiring – i.e. everybody has to adopt standard "voltage", "frequency" etc... This can/will be an issue (including in MITgcm) because right now there is no adopted standard for this wiring.
- Very appealing concept – not clear broader developer/user community "get it" yet!!!

- Strategy is OO programming based but....
  - Is designed to be compatible with common Fortran codes.
  - Draws very heavily on formal hardware design concepts VHDL ➔ Bluespec, etc....
- Future developments (maybe)
  - Parallel Python bindings.



```
 from pyclimate.svdeofs import *
 from pyclimate.ncstruct import *
   :
# Execute ESMF/MAPL component using PyMAPL interface
# (code not shown)
   :
# Do principle component and empirical orthogonal function analysis
# on computed field 'slp_djf'.
  slp = gsc.Export.GetPointer('slp_djf');
  pcs, lambdas, eofs = svdeofs( slp )
   :
```

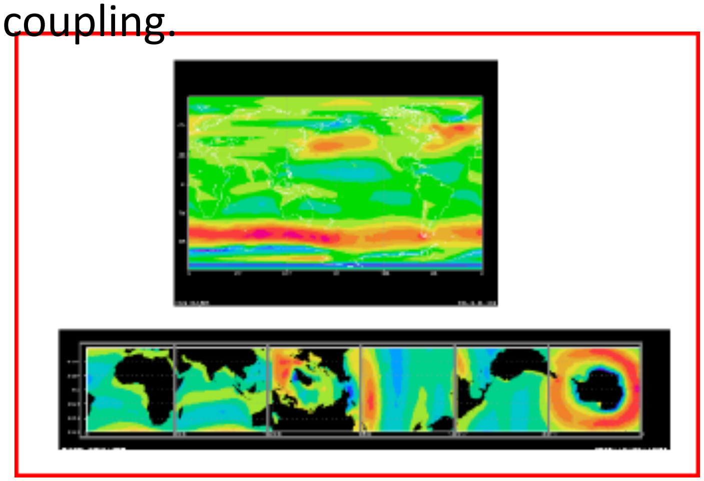
Listing 5: Hypothetical Python driver script connecting ESMF/MA

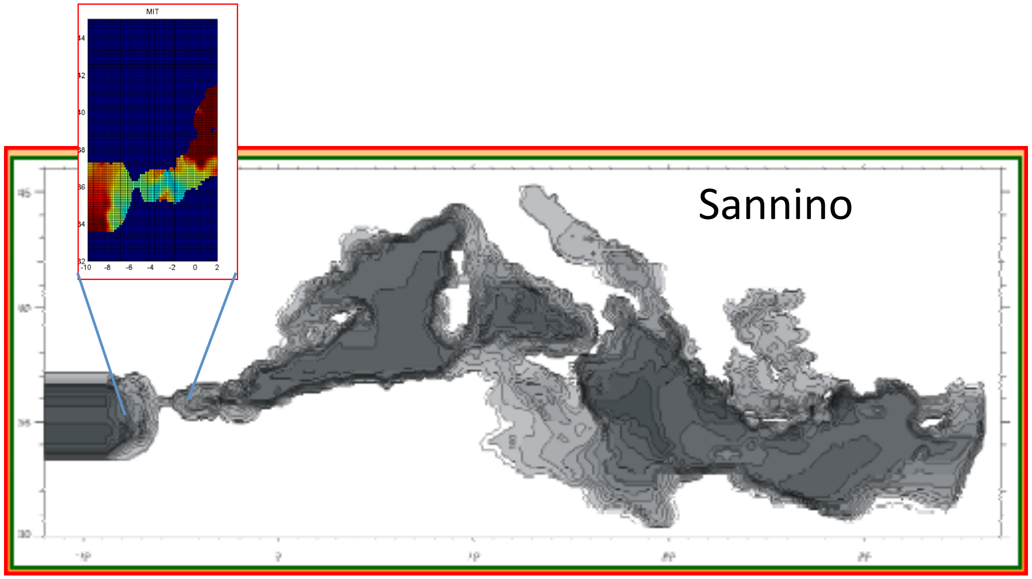
Coupling and regridding

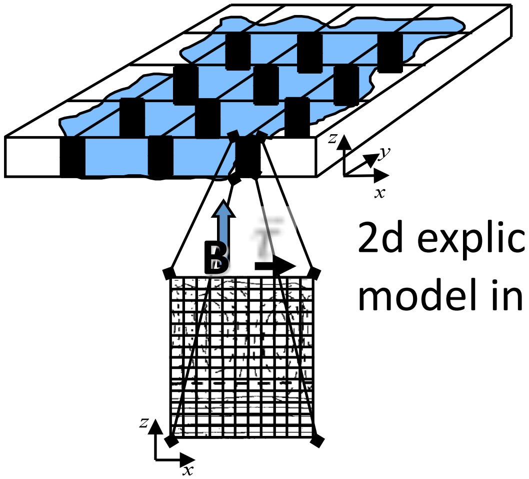
- Practical issue in multi-scale modeling is mapping between grids subject to constraints.
  - e.g. Local/global conservation
  - C0,C1,C2 continuous
  - Different coordinate bases.
  - Smoothing v. fitting
- State-of-the-art for this with parallel computations on the fly has historically been unsatisfactory
  - Patchwork of methods and software that work for regular lat-lon but can not work with less regular grids
  - Makes conceptually easy ideas hard to realize in practice

Coupling and regridding Examples

GEOS-5 lat-lon to MITgcm cube coupling.







2d explicit models embedded in 3d model in place of parameterization.

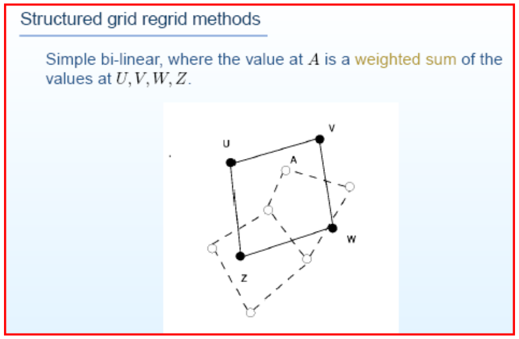
All examples involve two-way coupling across time and space scales, with constraints on interpolated fields e.g. local conservation, smooth curl field etc…

Currently no “unifying” toolset for this.

Other examples include biogeo, ice etc…

Generalized Coupling and regridding

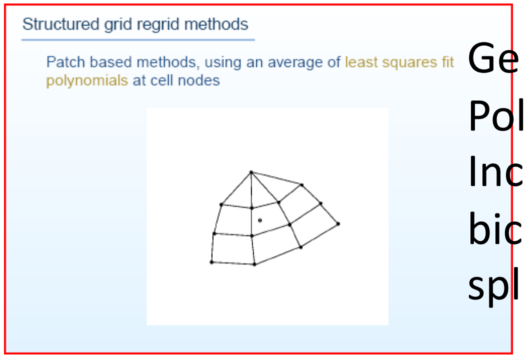
- Recent introduction into ESMF (after ~7 years!) – D. Neckels
- Compatible with MAPL
- Appears to work!



Structured grid regrid methods

Simple bi-linear, where the value at A is a weighted sum of the values at U, V, W, Z.

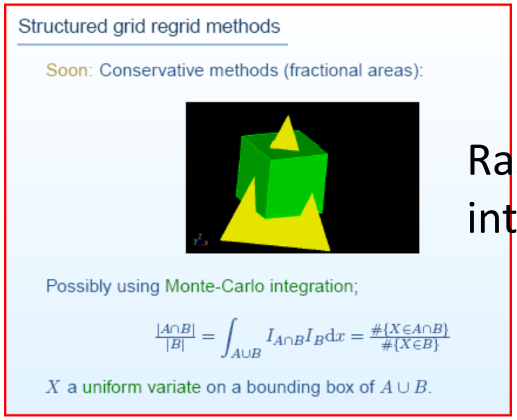
Simple bilinear



Structured grid regrid methods

Patch based methods, using an average of least squares fit polynomials at cell nodes

Generalized Polynomial – Includes bicubic, b-splne etc…



Structured grid regrid methods

Soon: Conservative methods (fractional areas):

Possibly using Monte-Carlo integration;

$$\frac{|A \cap B|}{|B|} = \int_{A \cup B} I_{A \cap B} I_B \mathrm{d}x = \frac{\#\{X \in A \cap B\}}{\#\{X \in B\}}$$

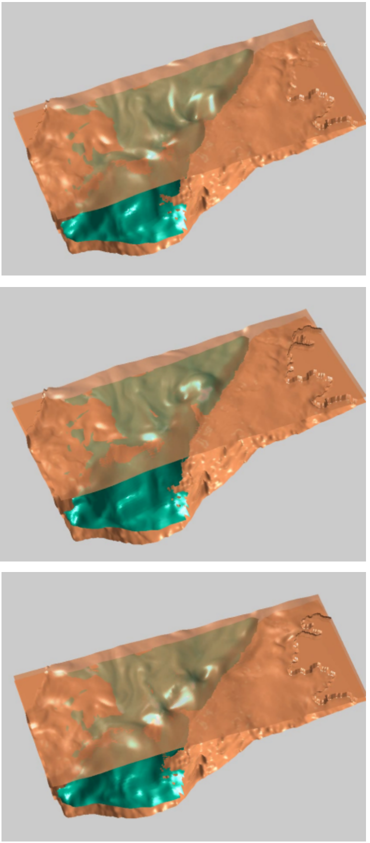
X a uniform variate on a bounding box of $A \cup B$.

Rasterized integration.

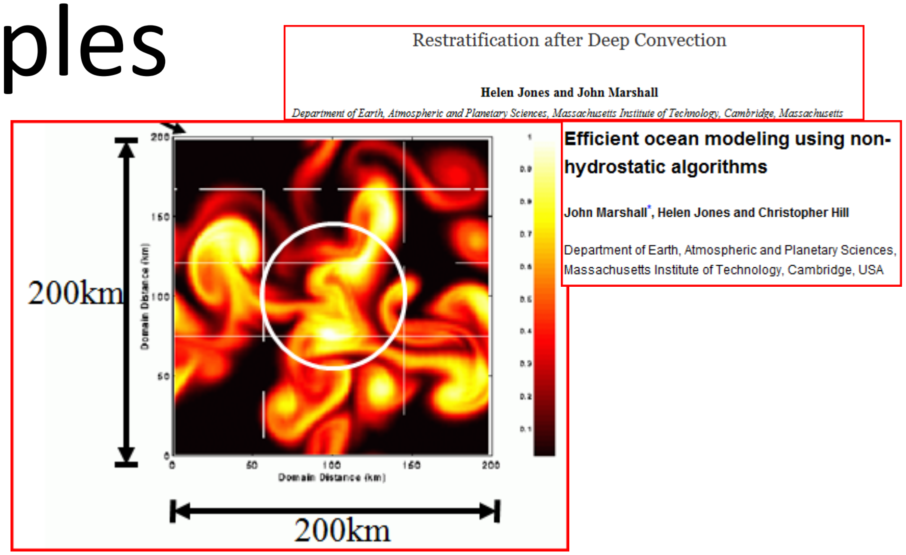
Summary

- Combination of conformal grid generation, MAPL and ESMF regridding have potential for interesting
  - High-res. Down to 1-2km.
  - Multi-scale. Embedding/ nesting with flexibility on conservation rules.
  - Multi-physics. Atmos/ ocean, ice/ocean, eco etc…
- Longer term we should revisit non-orthogonal grids.

Some deformation resolving MITgcm examples



T. Haine, JHU



Restratification after Deep Convection

Helen Jones and John Marshall

Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts

Efficient ocean modeling using non-hydrostatic algorithms

John Marshall[*], Helen Jones and Christopher Hill

Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, USA

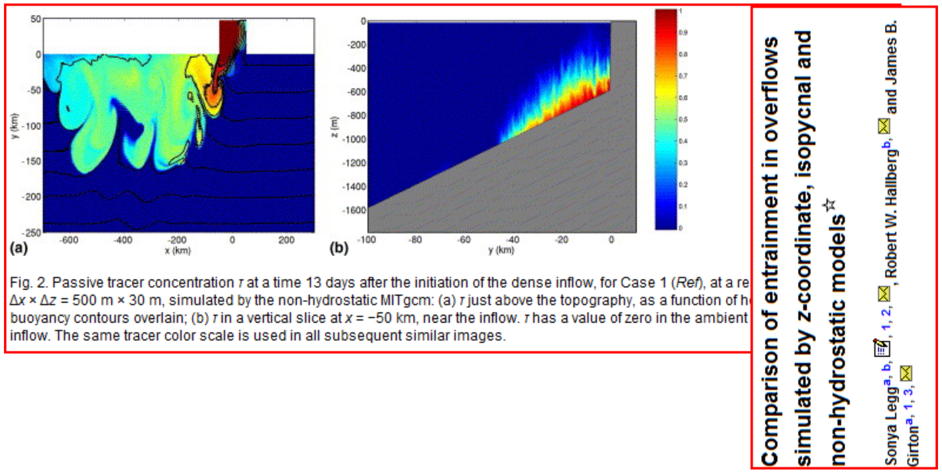


Fig. 2. Passive tracer concentration τ at a time 13 days after the initiation of the dense inflow, for Case 1 (*Ref*), at a re $\Delta x \times \Delta z$ = 500 m × 30 m, simulated by the non-hydrostatic MITgcm: (a) τ just above the topography, as a function of h buoyancy contours overlain; (b) τ in a vertical slice at $x = -50$ km, near the inflow. τ has a value of zero in the ambient inflow. The same tracer color scale is used in all subsequent similar images.

Comparison of entrainment in overflows simulated by z-coordinate, isopycnal and non-hydrostatic models☆

Sonya Legg[a, b], [1, 2], Robert W. Hallberg[b], and James B. Girton[a, 1, 3]

These as well as other studies consistently find qualitative improvements in accuracy @ Δ_h ≈ 2km or less